Accessing the Rako Bridge from an external application.

Introduction

The Rako Bridge can be used to control the installation from an external device connected to either the Ethernet or the RS232 port.

RS232

Product:

WRA-232

The connection to the RS232 port is via a 3 pin connector. The port is configured 9600,8,N,1
(9600 Baud, 8 Bits, No Parity, 1 Stop Bit). The protocol is the same as the RAV232/RAV232+ products and can be found in the document, 'Rako RS232 Command Summary'.
https://www.rakocontrols.com/media/1286/rako-rs232-command-summary.pdf

TCP/IP

In the following examples the NetBios name for the bridge is assumed to be 'rakobridge' (the default). This name can be changed using the web interface or the IP address of the bridge can be used.

Telnet

Products:

RA/RTC/WA/WTC-Bridge
APR/WRE-Bridge
WRA-232 (No Feedback)

A Telnet type interface is available on port 9761. This does not require a login. The protocol is the same as the 'Rako RS232 Command Summary'. This interface is used by Rasoft so CANNOT be used at the same time.

The interface can be used by typing: telnet rakobridge 9761

HTTP

Products:

RA/RTC/WA/WTC-Bridge
APR/WRE-Bridge

Requests can be made via the web interface by submitting a HTTP GET request in the following format:

http://rakobridge/rako.cgi?room=5&ch=4&com=3

The example would send Scene 1 command to room 5, channel 4.

If channel is omitted it will default to 0 meaning all channels.

A page containing the phrase "Success!" is returned.
From version 1.1.7:

Level can be set using: lev=0 to lev=255.
Events can be enabled/disabled: event=1&active=1 or active=0

XML

Products:

RA/RTC/WA/WTC-Bridge

An XML file can be downloaded which contains information about the current installation.
NOTE: the information first requires uploading to the Bridge from Rasoft.

http://rakobridge/rako.xml

An example of part of the XML file returned is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<rako>
  <info>
    <version>2.3.2 RA</version>
    <buildDate>Aug 09 2016 11:32:27</buildDate>
    <hostName>RAKOBRIDGE</hostName>
```

```
    <hostIP>192.168.25.64</hostIP>
    <hostMAC>00:04:A3:40:0B:85</hostMAC>
    <hwStatus>25</hwStatus>
    <dbVersion>-16</dbVersion>
  </info>
  <config>
    <requirepassword />
    <passhash>NAN</passhash>
    <charset>UTF-8</charset>
  </config>
  <rooms>
    <Room id="9">
      <Type>Lights</Type>
      <Title>Kitchen</Title>
      <mode>4+OFF</mode>
      <Scene id="1">
        <Name>Cooking</Name>
      </Scene>
      <Channel id="1">
        <type>Default</type>
        <Name>Downlights</Name>
        <Levels>FFBF7F3F000000000000000000000000</Levels>
      </Channel>
      <Channel id="2">
        <type>Default</type>
        <Name>Under cabinet</Name>
        <Levels>FFBF7F3F000000000000000000000000</Levels>
      </Channel>
    </Room>
  </rooms>
</rako>
```

Scene Cache

Products:

RA/RTC/WA/WTC-Bridge
APR/WRE-Bridge (V1.2.6+)

The bridge caches the scene state of up to 64 rooms. The cache indicates the last scene set. If the fade buttons are used in the room the entry is deleted from the cache. The scene cache can be obtained from the following URL
http://rakobridge/scenes.htm
The reply is 2 byte per room in hexadecimal:

Byte 1								Byte 2							
B7	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**	**B7**	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**
Scene Number					Room Number										

e.g `0x04041006` is room 4 scene 1 & room 6 scene 4
Note: The cache information can also be obtained by UDP.

The two byte format is the same as the UDP byte format in the next section.

UDP/IP

Products:

RA/RTC/WA/WTC-Bridge
APR/WRE-Bridge (V1.7.9 Limited support)

UDP Discover

All text is encoded with WINDOWS-1252

To find a bridge on the network send a UDP broadcast packet where the data consists of a single literal 'D'. The bridge will reply to the source IP address with:

Direction	Parameters			Example
	Byte	**Bit**	**Function**	
Client to UDP Broadcast	0	0:7	`0x44`: 'D' for request	`0x44`

The reply is send as a String:

Direction	Broadcast Reply From Bridge
Bridge To Client	NETBIOS_NAME\r\n Bridge MAC Address FF:FF:FF:FF:FF:FF \r\n

Scene Cache

This command requests data from the bridge it can be used to request stored level per scene data and scene cache.

Level per scene data is uploaded from Rasoft and when scenes are stored on the Smartphone apps (Persistent).

Scene Cache data is tracked from Bridge power up (Volatile). This data holds the last used scene for a particular room.

There is no current way of recalling the dimmers level but using the level cache and scene cache together can produce a good approximation.

Request Command

Direction	Parameters			Example
	Byte	Bit	Function	

Client to Bridge	0	0:7	`0x51`: 'Q' for query	`0x5101`
				Request Scene Cache
				
				`0x5121`
				Request Level and Scene Cache

	1	0:7	Type of request `0x01`: REQ_SCENECAHCE `0x20`: REQ_LEVEL `0x21`: REQ_SCENECAHCE + REQ_LEVEL *Note Other bits are used but not part of the API*	

Request replies one of the following:

Scene Cache Reply

Direction	Parameters			Example
	Byte	Bit	Function	

Bridge To Client	0	0:7	`0x43`: 'C' for cache	`0x4301FF`
				No items Cached
				
				`0x43030C1CD5`
				Room 28 Scene 3
				
				`0x43050C1C040CC3`
				Room 28 Scene 3
				Room 12 Scene 1

	1	3:7	Scene number *First 5 bits of byte 1*	
	1:2	0:10	Room number *Last 10 bits of byte 1&2*	
	...4:3	0:15	Repeat of two bytes above for amount of rooms (Maximum of 32 cached items) *Note cache can be empty and return only 2 bytes*	
	Last	0:7	CRC Byte position depending on amount of cached items	

Level Cache Reply

Direction	Parameters			Example
	Byte	Bit	Function	

Bridge To Client	0	0:7	`0x58`: 'X' for data record	`0x58`
				`04 80 49 01 ff bf 7f 3f 00 00 00 00 00 00 00 00 00 00 00`
				`04 80 49 02 ff bf 7f 3f 00 00 00 00 00 00 00 00 00 00 00`
				Room 73 Channel 1,2
				Text has been split into records

	1	0:7	Data Record Type `0x04`: Level cache reply	
	2	7	Active	
	2	6	Deleted	
	2	2:5	Reserved	
	2:3	0:9	Room	
	4	0:15	Channel	
	3	0:7	Scene 1 Level	
	...	...	...Continued from Byte 4-19	
	20	0:7	Scene 17 Level	
	20+		Repeat byte 1-20 for amount of records	
	Last	0:7	CRC Byte position depending on amount of cached items	

EOF Reply

Direction	Parameters			Example
	Byte	Bit	Function	
Bridge To Client	0	0:7	`0x58`: 'X' for data record	`0x4301FF` No items Cached
	1	0:7	Data Record Type `0xFF`: Level cache reply	
	2:3	0:15	Extended Checksum	
	4	0:7	CRC Byte position depending on amount of cached items	

Sending Commands

UDP packets can be sent to port 9761 with the following format:

Direction	Parameters			Example
	Byte	Bit	Function	

Client to Bridge	0	0:7	`0x52`: 'R' for request	`0x5207000400310105BE`
				Room 4, Channel 0
				Scene 5
				
				`0x5208000f0034010x7F0035`
				Room 15, Channel 0
				Level 50%

	1	0:7	Bytes to follow *5+data length*	
	2:3	0:9	Room	
	4	0:7	Channel	
	5	0:7	Command	
	6..x	0:7	Data[0..6] (Data Byte Count Min:0, Max:7)	
	Last	0:7	Checksum	

Refer to ‘Command List’ table for more information.

The reply is either "AOK" or "AERROR"

Status Messages

From version 1.2.2, status messages are sent out using UDP broadcast on port 9761. The format of the status messages is the same as the command packets except it starts with `0x53` and the CRC does not include the bytes-to-follow byte.

Direction	Parameters			Example
	Byte	**Bit**	**Function**	

Bridge to Client	0	0:7	`0x53`: 'S' for status	`0x530A00050031 0104000000C5`
				
				Room 5, Channel 0 Scene 4
				
				`0x530A0064003101 0300000067`
				
				Room 100, Channel 0 Scene 3

	1	0:7	Bytes to follow *5+data length*	
	2:3	0:9	Room	
	4	0:7	Channel	
	5	0:7	Command	
	6..x	0:7	Data[0..7] (Data Byte Count Min:0, Max:8)	
	Last	0:7	CRC (Checksum)	

UDP Instructions

Below is a table of available instructions for sending and for status messages. This is not the full extent of commands if you are trying to implement undefined commands it is recommended to contact us.

Instructions

Command	Name	Description	Data 0	Data 1
0x00	**OFF**			
0x01	**FADE_UP**	Fade the circuit in direction. Must be stopped with STOP (Release event)		
0x02	**FADE_DOWN**			
0x03	**SC1**	Legacy use SET_SCENE (This command appears in feedback so recommended to monitor)		
0x04	**SC2**			
0x05	**SC3**			
0x06	**SC4**			
0x08	**IDENT**	Causes the circuit to pulse or LED to flash in Ident pattern		
0x0C	**LEVEL_SET**	Legacy use SET_LEVEL Both data bytes must be level value (This command appears in feedback so recommended to monitor)	Level	Level
0x0D	**STORE**	When a keypad has finished saving a scene		
0x0F	**STOP**	Stop Fading		
0x2D	**CUSTOM_232**	Trigger a specific Custom String	Flags	String Id
0x2F	**HOLIDAY**	Set holiday mode (Refer to Flags table)	Flags	
0x31	**SET_SCENE**	Important:: Flags should be set to ‘Use Default Rate’	Flags	Scene#
0x32	**FADE**		Flags	Level
0x34	**SET_LEVEL**		Flags	Level

Flags

Command	Name	Data 0	Bit	Description
0x2D	**CUSTOM_232**	Flags	0:7	Reserved
0x2F	**HOLIDAY**	Flags	0:1	0x00: STOP_PLAYBACK 0x01: START_PLAYBACK 0x02: START_RECORD 0x03: STOP_RECORD
0x31	**SET_SCENE**	Flags	0	Use Default Fade Rate
			1:7	Reserved
0x32	**FADE**	Flags	0	Fade Up (false) / Fade Down (true)
			1:6	Reserved
			7	Use Default Fade Rate
0x34	**SET_LEVEL**	Flags	0	Use Default Fade Rate
			1:6	Reserved

Any Reserved values should be set to 0

Example UDP Commands

Room 7, Channel 0, Fade Up

Byte	Value	Description
0	0x52	Character ‘R’ for main commands
1	0x05	5 Bytes after this
2	0x00	Room Number High (Used for rooms above 256)
3	0x07	Room Number Low (Room 7)
4	0x00	Channel 0
5	0x01	FADE_UP Instruction
6	0xF3	Checksum (256 - (0x05 + 0x00 + 0x07 + 0x00 + 0x01) % 256)

Room 7, Channel 0, Scene 5

Byte	Value	Description
0	0x52	Character ‘R’ for main commands
1	0x07	7 Bytes after this
2	0x00	Room Number High (Used for rooms above 256)
3	0x07	Room Number Low (Room 7)
4	0x00	Channel 0
5	0x31	SET_SCENE Instruction
6	0x01	Flags ‘Use Default Rate’
7	0x05	Scene 5
8	0xBB	Checksum

Room 277, Channel 1, Level 64%

Byte	Value	Description
0	0x52	Character ‘R’ for main commands
1	0x07	7 Bytes after this
2	0x01	High Room number (room / 256)
3	0x15	Low Room number ((room % 256) + High Room)
4	0x01	Channel 1
5	0x0C	CAN_LEVEL Instruction
6	0x01	Flags ‘Use Default Rate’
7	0xA3	64% (64 * 2.55)
8	0x32	Checksum